National Aeronautics and Space Administration

IT Labs Proof-of-Concept Project: "Cloudbursting" Project

Office of the CTO for IT

Office of the Chief Information Officer

NASA IT Vision: The NASA IT Organization is the **very best** in government

George Chang, et al.

1/7/2015

Overview

Problem definition

- Center resources are stove piped, currently shared in a limited and difficult way requiring users to access via VPN. This environment inhibits collaborative science that requires the participation of multiple NASA centers.

Background

- Cloudbursting is model that builds on converged infrastructure and shared services. It enables on-demand access to resources across NASA's multiple administrative domains in a secure manner. and thus maximizes the effectiveness of shared resources. We believe that this technology will bring down the barriers to the resource sharing problem and enhance the effectiveness of shared resources in achieving NASA missions.

Expected Project Outcome:

- Demonstration of the feasibility of cloudbursting of computational load across cloud clusters administered by ARC, JPL and LaRC.

Alignment with 2011 NASA IRM Strategic Goals and Objectives

- **Goal 1: Transform NASA's IT infrastructure and application services.**

  Objective 1.5 – Enhance mission success by providing efficient and effective access to enterprise information and collaborative functionality.

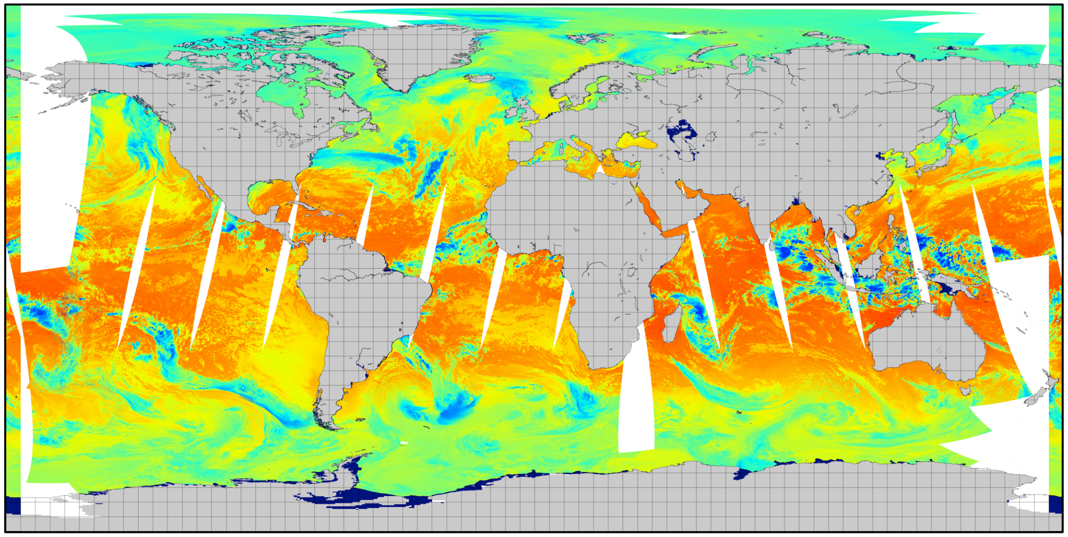
  » A multi-center infrastructure that supports efficient and secure cloudbursting will enable on-demand access to resources and maximize the effectiveness of shared resources. It will support an environment for NASA scientists and engineers to collaborate on missions by pooling the computational resources.

Approach

- Demonstrate the capability and feasibility through a simple use case
  - » Set up cloud cluster at each center
  - » Build use case virtual machine
  - » Test cloudbursting across centers
- Identify the issues which inhibit cloudbursting amongst NASA centers
- Document the findings and recommendations

Use Case: Image Generation

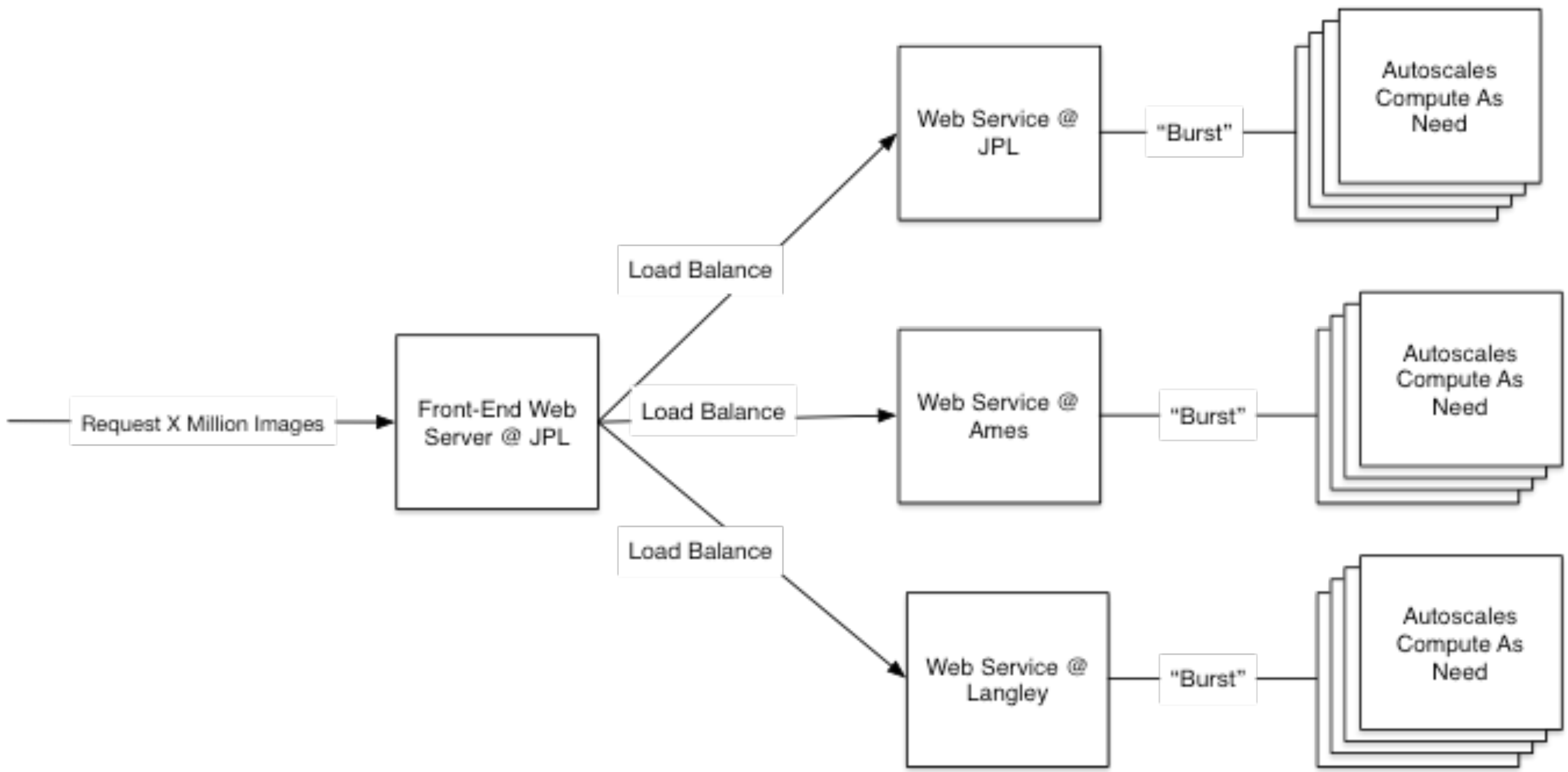
- We selected a use case that is relevant to Earth Science research. A typical operation performed at the NASA's Physical Oceanography Distributed Active Archive Center (PO.DAAC) is generation of Level 3 imagery from Level 2 data products.



Methodology

1. Configure local (JPL) Eucalyptus instance using the faststart image.
2. Replicate the configurations at the other centers to make the installations consistent between centers.
3. Obtain the GHRSST Level 2P Global Skin Sea Surface Temperature from Moderate Resolution Imaging Spectroradiometer (MODIS) on the NASA Aqua satellite
4. Develop virtual machine instance that performs data-to-image workflow operation and verify single center (JPL) operations.
5. Expand image generation to “burst” computing to one other datacenter.
6. Investigate auto-scaling capability to support daisy chain bursting, that is, execute job at JPL, burst to ARC, and burst to LaRC.
7. Gather performance statistics, which includes throughputs when processing in a single datacenter, and bursting to other datacenters.

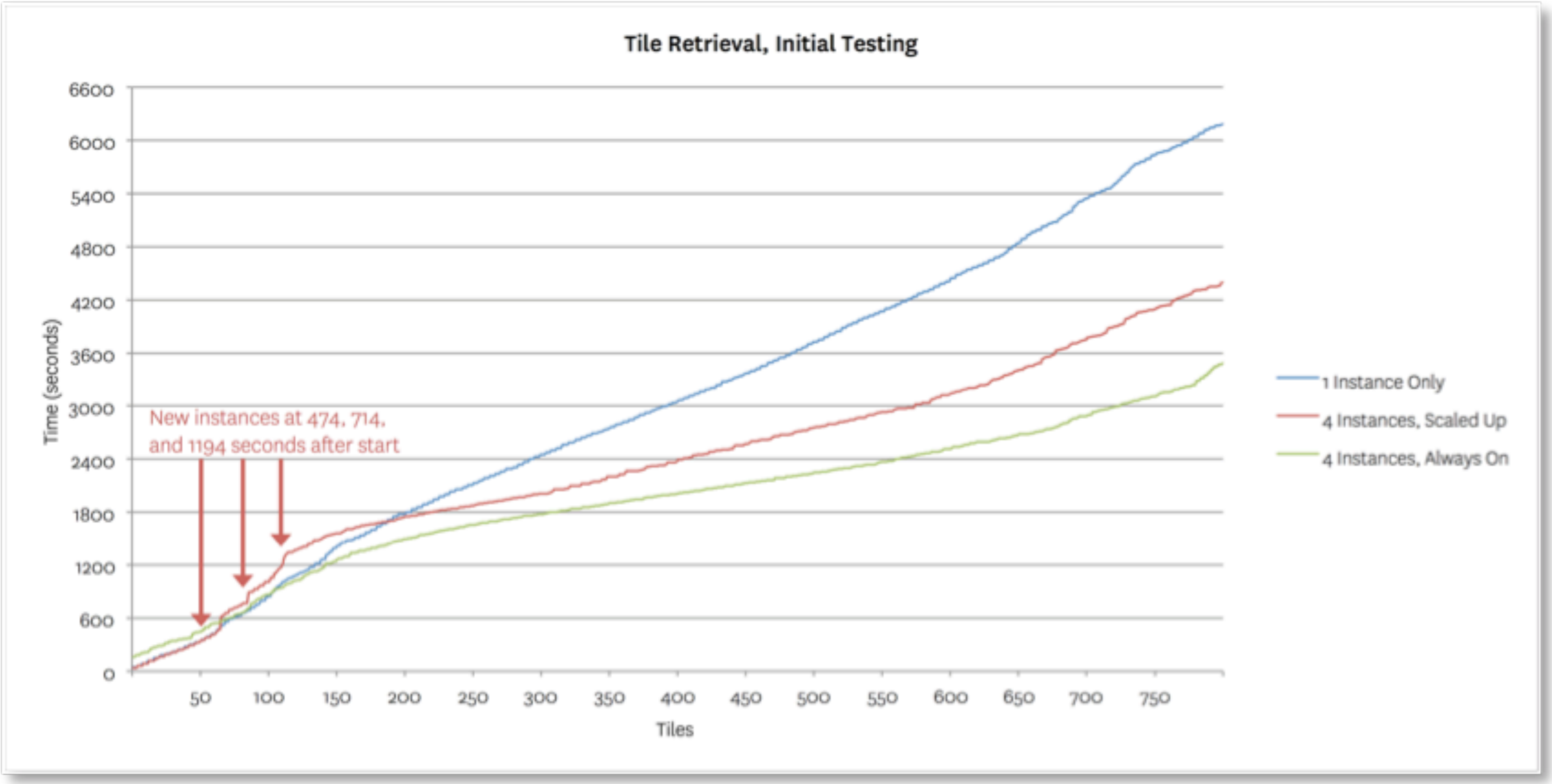
Architecture



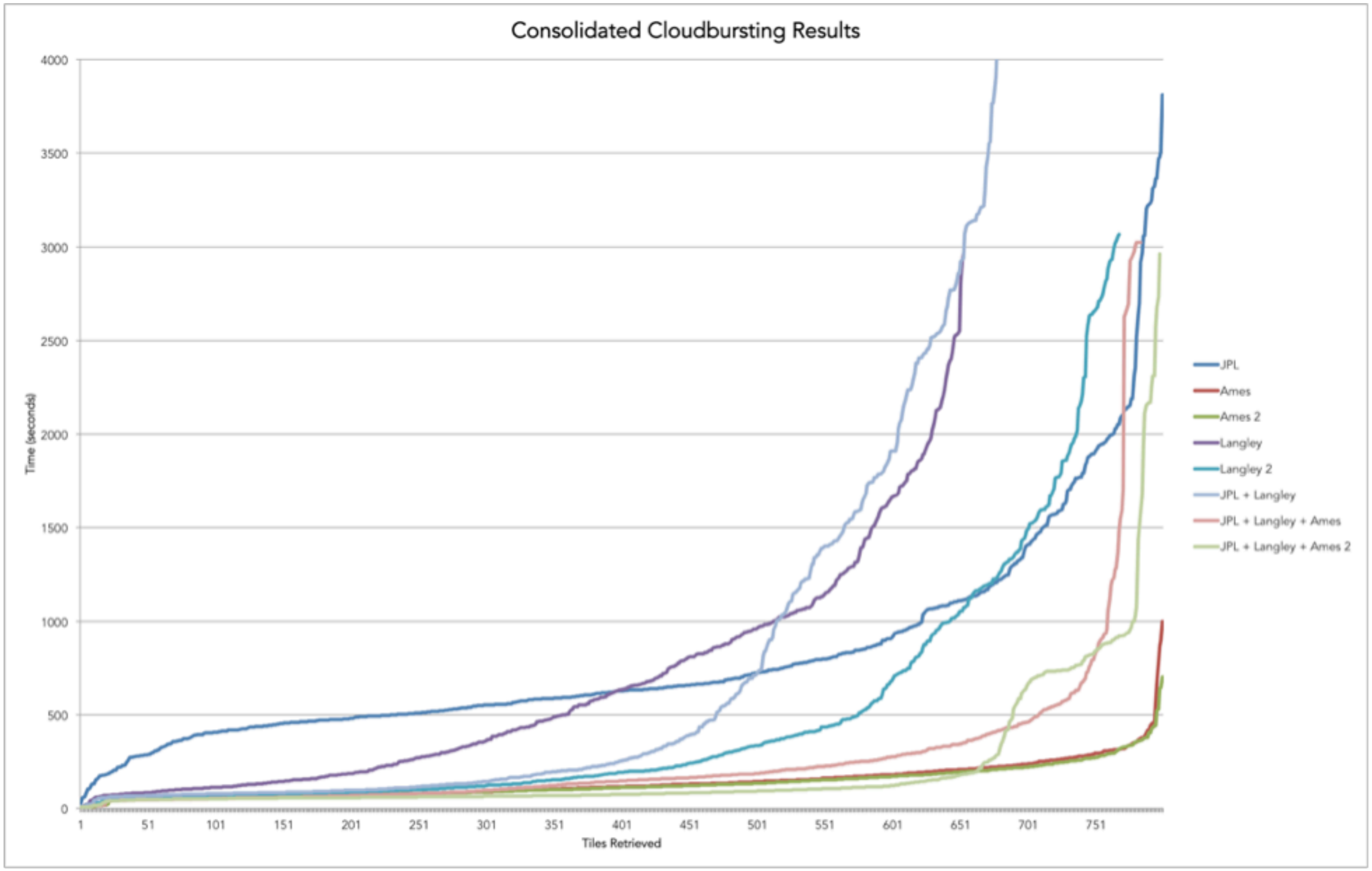
Hardware

- **JPL**
  - 1 Primary System – Eucalyptus Frontend
  - 12 Core Intel Xeon E5 w/ 24GB RAM, 500GB PCIe SSD Storage
  - 4 Eucalyptus Nodes, each with 2 virtualized CPU and 2GB RAM
  - Eucalyptus software stack version: 3.4.2 on CentOS 6.5
- **Ames**
  - 16 Nodes with dual 8-core Sandy Bridge chips (16 cores per node), 64 GB of ram
  - Disks: there are 2 x 1T disks in each node
    - Allocated to the OS, some part for OpenStack images storage and the rest of the disk is available as temp "local" storage for any instance running on that node.
    - The other disk is part of a shared global OpenStack file system. An EBS-like storage but under OpenStack.
  - Interconnect: dual 1GigE Ethernet between the nodes and the front end
  - Software stack: Eucalyptus 4.0 on CentOS 6.5
- **LaRC**
  - 1 Primary System (Eucalyptus "Frontend")
    Eucalyptus Components: CLC (Cluster Controller) and CC (Cloud Controller)
    Eucalyptus Software stack version: Eucalyptus 3.4.3
  - 1 Storage Controllers (Eucalyptus "Nodes")
    Eucalyptus Components: SC (Storage Controller) and Walrus (S3) server
  - 2 Node Controllers (Eucalyptus "Nodes")
    Eucalyptus Components: NC (Node Controller)

Single-Center Results



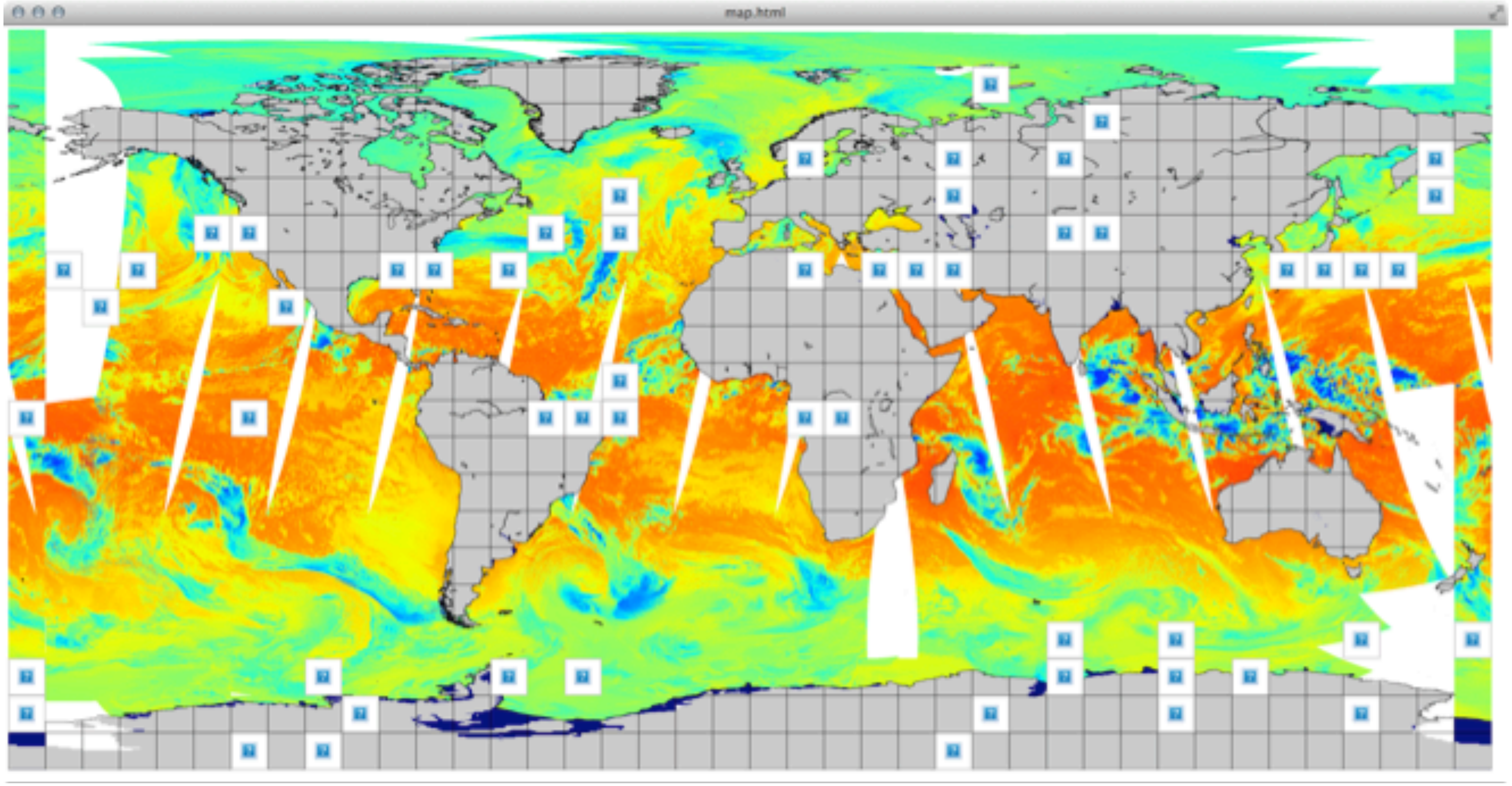
Multi-Center Results



Test Results

- **JPL**
  - » Observation: Running Eucalyptus 3.4.2 with predictable results.
  - » Autoscaling: Using internal load balancing properly adds and removes instances depending on the overall load of the cluster.
  - » Cluster would become unstable after prolonged uptime. Suspicion is with IT security randomly running vulnerability tests that caused the instances to become loaded and autoscale seemly randomly.
- **ARC**
  - » Observation: Running Eucalyptus 4.0 had difficulties with internal load balancing and set up due to incomplete documentation at the time.
  - » Autoscaling: Initial scaling up works well, but subsequent scaling back up after down scaling.
- **LaRC**
  - » Observation: Running Eucalyptus 3.4.2 at LaRC works well with internal load balancing.
  - » Autoscaling: Instances would frequently end up in a reboot cycle where instances would terminate only to be started up again.

Test Results



Issues/Challenges

- IT security is a major issue in enabling cross-center sharing of resources
  - » IP Address block allocation
  - » Firewall issues
  - » Network security
- Cloud platform Eucalyptus is easy to set up, but was difficult and time consuming to tailor for our specific use case
  - » Load balancer configuration
  - » Same installation, different parameters
- Not enough time to assess and learn more:
  - » How to set up a federated cloud environment?
  - » Can different cloud platforms be federated together seamlessly?
  - » How best to move data from one cloud to another?
  - » What is the best private cloud platform/vendor?
  - » What governance structure needs to be put in place?)

Findings/Observations (if applicable)

- Cloudbursting and computational resource sharing are feasible using current cloud frameworks, however, more effort is needed to establish a seamless environment that works across security domains
- Each center can manage the amount of resources and number of instances given to a particular project for sharing

Recommendations

- Extend the proof-of-concept to include other NASA cloud resources to study issues such as:
  - » Mechanisms to setup and manage a federated cloud environment
  - » Governance structure required for a federated environment
  - » Security issues in accessing resources across security domains
  - » Reliable and efficient mechanisms to move data and information across centers
- Develop proof-of-concept environment based on multiple cloud platforms
  - » Experiment with utilizing other platforms e.g., OpenStack
  - » Provide feedback to cloud platform developers on enhancement required to support cloudbursting in NASA
  - » Demonstrate a federated environment utilizing multiple cloud platforms

Project Team

- Project Manager: Mike Little/Langley
- Project Team (Civil Servant and Contractors):
  - » Andrei Vakhnin, John Koening/LaRC
  - » Piyush Mehrotra, Steve Heistand, Bob Hood/ARC
  - » Emily Law, George Chang, Thomas Huang, Hook Hua, Paul Zimdars/JPL